IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for scheduling program units, the method comprising:

starting a process within an operating system;

starting a plurality of program units within [[an]] the operating system, the program units associated with the process; and

upon the occurrence of a context shifting event for a first program unit of the plurality of program units, synchronizing the scheduling of each of the plurality of program units and setting the context of each of the plurality of program units [[to]] such that each of the program units process the same context shifting event as the first program unit.

2. (Original) The method of claim 1, wherein the program unit comprises a thread.

3. (Currently Amended) The method of claim 2, wherein ~~each of the threads are~~ at least one thread is executed on a separate processor from the processor executing at least one other thread.

4. (Original) The method of claim 3, wherein the processors executing the threads reside on at least a first multiple processor unit and a second multiple processor unit, and further comprising migrating threads executing on the second multiple processor unit to the first multiple processor unit.

5. (Original) The method of claim 1, wherein the context shifting event comprises an exception.

6. (Original) The method of claim 5 wherein the exception comprises a signal.

7. (Original) The method of claim 1 wherein the context shifting event comprises a non-local goto.

8. (Original) The method of claim 1, wherein the context shifting event comprises a system call.

9. (Currently Amended) A system for scheduling program units, the system comprising:
at least one multiple processor unit having a plurality of processors;
a memory coupled to the plurality of processors; and
an operating environment stored in the memory and executed by at least one of the processors ~~from the memory and~~ wherein at least one of the processors is operable to perform the tasks of:

start a process,

start a plurality of program units within an operating system, the program units associated with the process, and

upon the occurrence of a context shifting event for a first program unit of the plurality of program units, synchronize the scheduling of each of the plurality of program units and setting the context of each of the plurality of program units [[to]] such that the program units process the same context shifting event as the first program unit.

10. (Original) The system of claim 9, wherein the program unit comprises a thread.

11. (Currently Amended) The system of claim 10, wherein ~~each of the threads are~~ at least one thread is executed on a separate processor of the plurality of processors from the processor executing at least one other thread.

12. (Currently Amended) The system of claim 11, wherein the processors executing the threads reside on at least a first multiple processor unit and a second multiple processor unit ~~of the at least one multiple processor units~~, and wherein the operating environment migrates threads executing on the second multiple processor unit to the first multiple processor unit.

13. (Original) The system of claim 9, wherein the context shifting event comprises an exception.

14. (Original) The system of claim 13 wherein the exception comprises a signal.

15. (Original) The system of claim 9 wherein the context shifting event comprises a non-local goto.

16. (Original) The system of claim 9, wherein the context shifting event comprises a system call.

17. (Currently Amended) A tangible computer-readable media having computer-executable instructions for performing a method for scheduling program units, the method comprising:

starting a process within an operating system;

starting a plurality of program units within [[an]] the operating system, the program units associated with the process; and

upon the occurrence of a context shifting event for a first program unit of the plurality of program units, synchronizing the scheduling of each of the plurality of program units and setting the context of each of the plurality of program units [[to]] such that the program units process the same context shifting event as the first program unit.

18. (Currently Amended) The tangible computer-readable media of claim 17, wherein the program unit comprises a thread.

19. (Currently Amended) The tangible computer-readable media of claim 18, wherein ~~each of the threads are~~ at least one thread is executed on a separate processor from at least one other thread.

20. (Currently Amended) The tangible computer-readable media of claim 19, wherein the processors executing the threads reside on at least a first multiple processor unit and a second multiple processor unit, and wherein the method further comprises migrating threads executing on the second multiple processor unit to the first multiple processor unit.

21. (Currently Amended) The tangible computer-readable media of claim 17, wherein the context shifting event comprises an exception.

22. (Currently Amended) The tangible computer-readable media of claim 21 wherein the exception comprises a signal.

23. (Currently Amended) The tangible computer-readable media of claim 17 wherein the context shifting event comprises a non-local goto.

24. (Currently Amended) The tangible computer-readable media of claim 17, wherein the context shifting event comprises a system call.